REMARKS

Claims 1-5, 7-11 and 13-23 are pending in the present application. Claims 6 and 12 were canceled in a previous amendment. Applicant respectfully submits that each of these claims is allowable without amendment. Therefore, Applicant respectfully requests reconsideration of the claims in view of the following remarks.

Applicant has provided a replacement sheet for Figure 3, where the step 15 has been corrected to state "Form H_n," as opposed to the previously incorrect "Form H_0." The drawing is now consistent with Paragraph [0039] of the specification as published. "If the answer is 'yes' the algorithm proceeds to step 15 in which … the CRC is used to generate look-up table address H_n from A_n." Par. [0039].

Claims 1-5, 8-17 and 19-23 have been rejected under 35 U.S.C. § 102(b) as being anticipated by Kaku (U.S. Patent No. 6,279,097). Claims 7 and 18 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Kaku. Applicant respectfully traverses these rejections.

Claim 1, as previously presented, specifically recites "using A_0 to generate y+1 look-up table addresses H_0, H_1, H_2,...,H_y, where y is an integer greater than or equal to one, wherein each of the addresses H_1, H_2, …, H_y is obtained from the address A_0 by first forming a respective string A_n having the same number of bits as A_0, and then applying the algorithm by which H_0 is obtained from A_0." The references of record do not teach or suggest the limitations of claim 1.

In particular, Kaku never teaches or suggests forming a string A_n having the same number of bits as a MAC address A_0 and using this string to generate a lookup table address using the same algorithm that was applied to the MAC address to generate the first lookup table address. As shown in Figure 1, the table addresses are generated by Barrel Shifter 26 and Bucket Index

Pointer (BIXP) 40. The barrel shifter 26 receives its input from CRC32 Generator 24, which generates a compressed address which has fewer bits than the input address. Col. 4, line 29. Kaku never teaches or suggest applying the algorithm by which H_0 is obtained from A_0 to a string A_n that has the same number of bits as A_0.

The Examiner responded to the arguments Applicant made in the earlier amendment. These will be addressed now.

[A] The Examiner notes that Kaku teaches (Col. 5, lines 21-30) that two registers exist that have two data strings in them. Applicant agrees that these registers exist and each stores a 48-bit address along with 16 control bits. Neither of these registers, however, stores an address that performs the function of the claimed An. In particular, the addresses H_1, H_2, …, H_y are obtained from the address A_0 by first forming a respective string A_n and then applying the algorithm by which H_0 is obtained from A_0. Kaku's registers 36 and 38, on the other, merely store the input address AI, i.e., the contents of the lookup table. These registers have no bearing on generating addresses for the lookup table.

[B] The Examiner also addresses Applicant's argument that Kaku does not generate a second lookup table address by generating a string having the same number of bits as the MAC address. In this argument, the Examiner states that the string is formed at the address output register, which has 48 bits going to the address comparator. Applicant respectfully disagrees with this analysis.

The address output register 38 never stores bits that are used to generate a lookup address by applying the algorithm by which H_0 is obtained from A_0, as required by claim 1. Applicant agrees that the address output register 38 provides 48 bits to the address comparator 32. In state 72 of the flowchart of Figure 5, the address comparator 32 compares the 48-bit input address

with the 48-bit address in the address output register that was read from the address lookup table 28. Col. 7, line 45. This check is performed to see if the input address has already been stored. If it has, the control state machine returns to the idle state 34. If the addresses do not match, the algorithm continues. At no point, however, does the algorithm ever generate a lookup address from the contents of the address output register, much less generate this address in the manner required by claim 1.

The Examiner finally notes that if the intent of Applicant is to claim the generation of lookup addresses from a string that is the same size but non-identical to the MAC address from which it was generated then this language should be in the claim. According to claim 1, look-up table addresses H_0, H_1, H_2,...,H_y are obtained from the address A_0 by first forming a respective string A_n and then applying the algorithm by which H_0 is obtained from A_0. Assume for the purpose of this discussion that A_n is identical to A_0. In that case, applying the algorithm by which H_0 is obtained from A_0 to A_n will necessarily lead to H_0. In other words, under this scenario the look-up table addresses H_0, H_1, H_2,...,H_y would all be H_0 (i.e., H_0, H_0, H_0,...,H_0). Certainly, Kaku never teaches or suggests generating look-up table addresses that are all the same. As a result, the suggested amendment is not necessary to overcome the art rejection.

[C] Finally, the Examiner notes that Applicant has not challenged or traversed the official notice taken in the previous office action. Applicant admits that Walsh codes were known prior to Applicant's 2002 priority date. This admission, however, should not be construed to an agreement that it would be obvious to modify the teachings of Kaku. This is certainly not the case. Given the allowability of the independent claims, however, we never reach this issue.

Claims 2-5, 7, 9 and 11 depend from claim 1 and add further limitations. It is respectfully submitted that these dependent claims are allowable by reason of depending from an allowable claim as well as for adding new limitations.

Claim 8 specifically recites "[a] switch including…a processor for associating MAC addresses with addresses of the look-up table." The processor is arranged "to use each MAC address A_0 to generate y+1 look-up table addresses H_0, H_1, H_2,..., H_y for y an integer greater than or equal to one, wherein each of the addresses H_1, H_2, …, H_y is obtained from the address A_0 by first forming a respective string A_n having the same number of bits as A_0, and then applying the algorithm by which H_0 is obtained from A_0, and according to at least one criterion to associate the address A_0 with a selected one of the addresses H_0, H_1, H_2,..., H_y." As discussed above with respect to claim 1, it is respectfully submitted that claim 8 is allowable over the references of record.

Claim 10 depends from claim 8 and adds further limitations. It is respectfully submitted that this dependent claim is allowable by reason of depending from an allowable claim as well as for adding new limitations.

Claim 13 specifically recites "generating a first look-up table address based upon the MAC address, the first look-up address being generated using an algorithm … [and] generating a second look-up table address by forming a string having the same number of bits as the MAC address and applying the algorithm to the string." Applicant respectfully submits that the references of record do not teach or suggest the limitations of claim 13.

In rejecting claim 13, the Office Action cites to Figs 5A-5B to indicate that if the first address is occupied, then a second/third/fourth address is generated using the string with the algorithm applied. Applicant respectfully submits that Kaku does not teach or suggest generating

a second look-up table address by forming a string having the same number of bits as the MAC address and applying the algorithm to the string.

Kaku teaches taking a 48-bit MAC address and compressing the 48-bit input address to generate a 32-bit compressed address. Col. 7, line 20; Fig. 5A, state 62. While this may be considered generating a first look-up table address based upon a MAC, Kaku never then generates a second look-up table address by forming a string having the same number of bits as the MAC address. In particular, Kaku never forms a string having 48 bits. As discussed above, the address output register merely stores an address that was previously stored in the look up table. No algorithm is ever applied to these bits. Therefore, it is respectfully submitted that claim 13 is allowable over the references of record.

Claims 14-21 depend from claim 13 and add further limitations. It is respectfully submitted that these dependent claims are allowable by reason of depending from an allowable claim as well as for adding new limitations.

Claim 22 specifically recites "generating a first look-up table address by applying an algorithm to the MAC address ... [and] if the first look-up table address is not associated with the MAC address, generating a second look-up table address by forming a string having the same number of bits as the MAC address and applying the algorithm to the string." As discussed with respect to claim 13, Kaku does not teach or suggest this limitation.

Claim 23 depends from claim 22 and adds further limitations. It is respectfully submitted that this dependent claim is allowable by reason of depending from an allowable claim as well as for adding new limitations.

Applicant has made a diligent effort to place the claims in condition for allowance. However, should there remain unresolved issues that require adverse action, it is respectfully

requested that the Examiner telephone Ira S. Matsil, Applicant's attorney, at 972-732-1001 so that such issues may be resolved as expeditiously as possible. No fee is believed due in connection with this filing. However, should one be deemed due, the Commissioner is hereby authorized to charge, or credit any overpayment, to Deposit Account No. 50-1065.

Respectfully submitted,

February 28, 2008
Date

/Ira S. Matsil/
Ira S. Matsil
Attorney for Applicant
Reg. No. 35,272

SLATER & MATSIL, L.L.P.
17950 Preston Rd., Suite 1000
Dallas, Texas 75252
Tel.: 972-732-1001
Fax: 972-732-9218